


```
FFFFFFFFF 000000 RRRRRRR RRRRRRR EEEEEEEEE WW WW IIIIII NN NN DDDDDDD
FFFFFFFFF 000000 RRRRRRR RRRRRRR EEEEEEEEE WW WW IIIIII NN NN DDDDDDD
FF 00 00 RR RR RR RR EE WW WW II NN NN DD DD
FF 00 00 RR RR RR RR EE WW WW II NN NN DD DD
FF 00 00 RR RR RR RR EE WW WW II NN NN DD DD
FFFFFFFF 00 00 RRRRRRR RRRRRRR EEEEEEE WW WW II NN NN DD DD
FFFFFFFF 00 00 RRRRRRR RRRRRRR EEEEEEE WW WW II NN NN DD DD
FF 00 00 RR RR RR RR EE WW WW II NN NN DD DD
FF 00 00 RR RR RR RR EE WW WW II NN NN DD DD
FF 00 00 RR RR RR RR EE WW WW II NN NN DD DD
FF 00 00 RR RR RR RR EE WW WW II NN NN DD DD
FF 000000 RR RR RR RR EEEEEEEEE WW WW IIIIII NN NN DDDDDDD
FF 000000 RR RR RR RR EEEEEEEEE WW WW IIIIII NN NN DDDDDDD

LL IIIIII SSSSSSS
LL IIIIII SSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LLLLLLLLL IIIIII SSSSSSS
LLLLLLLLL IIIIII SSSSSSS
```

```
1 0001 0 MODULE FOR$REWIND ( ! FORTRAN REWIND Statement
2 0002 0 IDENT = '1-007' ! File: FORREWIND.B32 ! Edit SBL1007
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
11 0011 1 * ALL RIGHTS RESERVED. *
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
18 0018 1 * TRANSFERRED. *
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
22 0022 1 * CORPORATION. *
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY: FORTRAN Support Library, user callable
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1 Contains routine FOR$REWIND: rewind a FORTRAN sequential
37 0037 1 access file.
38 0038 1
39 0039 1 ENVIRONMENT: Mixture of AST level or not.
40 0040 1
41 0041 1 AUTHOR: Jonathan M. Taylor, CREATION DATE: 10-OCT-77
42 0042 1
43 0043 1 MODIFIED BY:
44 0044 1
45 0045 1 Jonathan M. Taylor, 10-OCT-77 : VERSION 0
46 0046 1 Previous edit history removed. SBL 16-June-1982
47 0047 1 1-001 - Update version number and copyright notice. JBS 16-NOV-78
48 0048 1 1-002 - Change REQUIRE file names from FOR... to OTS... JBS 06-DEC-78
49 0049 1 1-003 - Change prefix of LUN literals from OPEN to LUB. JBS 13-DEC-78
50 0050 1 1-004 - Implement ERR= and IOSTAT=. SBL 1-May-1979
51 0051 1 1-005 - Error instead of no-op on not open or direct. SBL 2-May-1979
52 0052 1 1-006 - 1-005 is a mistake. No-op if not open, error if not
53 0053 1 sequential org and access. SBL 16-May-1979
54 0054 1 1-007 - Allow errors RMS$ IOP, RMS$ BOF and RMS$ EOF from $REWIND.
55 0055 1 Move declaration of ACTUALCOUNT. Add SWITCHES. SBL 16-June-1982
56 0056 1 --
```

```
58 0057 1 |
59 0058 1 | SWITCHES:
60 0059 1 |
61 0060 1 |
62 0061 1 | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
63 0062 1 |
64 0063 1 |
65 0064 1 | LINKAGES:
66 0065 1 |
67 0066 1 | REQUIRE 'RTLIN:OTSLNK';           ! Define all linkages
68 0495 1 |
69 0496 1 |
70 0497 1 | TABLE OF CONTENTS:
71 0498 1 |
72 0499 1 |
73 0500 1 | FORWARD ROUTINE
74 0501 1 |     FOR$REWIND;                   ! FORTRAN REWIND statement processor
75 0502 1 |
76 0503 1 |
77 0504 1 | INCLUDE FILES:
78 0505 1 |
79 0506 1 |
80 0507 1 | REQUIRE 'RTLML:FORERR';           ! FORTRAN error number definitions
81 0575 1 | REQUIRE 'RTLML:OTSLUB';           ! Logical Unit Block definitions
82 0715 1 | REQUIRE 'RTLIN:OTSMAC';           ! Macros
83 0909 1 | REQUIRE 'RTLIN:RTLPSECT';         ! Define DECLARE_PSECTS macro
84 1004 1 | REQUIRE 'RTLML:OTSISB';           ! ISB definitions
85 1172 1 | REQUIRE 'RTLML:FORPAR';           ! FORTRAN inter-module parameters
86 1195 1 |     LIBRARY 'RTLSTARLE';          ! STARLET library for macros and symbols
87 1196 1 |
88 1197 1 | MACROS:
89 1198 1 |     NONE
90 1199 1 |
91 1200 1 |
92 1201 1 | EQUATED SYMBOLS:
93 1202 1 |     NONE
94 1203 1 |
95 1204 1 |
96 1205 1 |
97 1206 1 | OWN STORAGE:
98 1207 1 |     NONE
99 1208 1 |
100 1209 1 |
101 1210 1 | EXTERNAL REFERENCES:
102 1211 1 |
103 1212 1 | EXTERNAL ROUTINE
104 1213 1 |     FOR$$IOSTAT_HND,              ! error condition handler
105 1214 1 |     FOR$$SIGNAL_STO: NOVALUE,     ! convert error number and signal
106 1215 1 |     FOR$$CB_PUSH: JSB_CB_PUSH NOVALUE, ! create LUB/ISB/RAB, if needed
107 1216 1 |     FOR$$CB_POP: JSB_CB_POP NOVALUE; ! return I/O system to previous state
108 1217 1 |
109 1218 1 |
110 1219 1 | PSECT DECLARATIONS:
111 1220 1 |
112 1221 1 |
113 1222 1 |     DECLARE_PSECTS (FOR);         ! declare PSECTS for FOR$ facility
```

```
115 1223 1 GLOBAL ROUTINE FOR$REWIND (  
116 1224 1     UNIT  
117 1225 1     ERR_EQL)  
118 1226 1     =  
119 1227 1  
120 1228 1 ++  
121 1229 1 FUNCTIONAL DESCRIPTION:  
122 1230 1  
123 1231 1     Perform RMS rewind operation on the file specified by the  
124 1232 1     UNIT parameter.  
125 1233 1  
126 1234 1 FORMAL PARAMETERS:  
127 1235 1  
128 1236 1     UNIT.rl.v      Logical unit number  
129 1237 1     ERR_EQL.rl.v  If 0 or not present, signal errors  
130 1238 1                  If non-zero, unwind to caller.  
131 1239 1  
132 1240 1 IMPLICIT INPUTS:  
133 1241 1  
134 1242 1     LUB$V_DIRECT      This unit has previously been specified  
135 1243 1                  for direct access by an OPEN statement or  
136 1244 1                  DEFINE FILE.  
137 1245 1     LUB$V_OPENED     This unit has already been opened by  
138 1246 1                  an OPEN statement or default open.  
139 1247 1  
140 1248 1 IMPLICIT OUTPUTS:  
141 1249 1  
142 1250 1     LUB$L_LOG_RECNO    set to 1.  
143 1251 1  
144 1252 1 ROUTINE VALUE:  
145 1253 1  
146 1254 1     An IOSTAT value.  
147 1255 1  
148 1256 1 SIDE EFFECTS:  
149 1257 1  
150 1258 1     SIGNAL STOPS FOR$ REWERR if a non-EOF error is returned from  
151 1259 1     the RMS rewind call.  
152 1260 1  
153 1261 1 --  
154 1262 1  
155 1263 2 BEGIN  
156 1264 2  
157 1265 2 GLOBAL REGISTER  
158 1266 2     CCB = 11: REF BLOCK[, BYTE];  
159 1267 2  
160 1268 2 LOCAL  
161 1269 2     STATUS,                ! Return status from $REWIND  
162 1270 2     L_UNWIND_ACTION: VOLATILE, ! Unwind action code (FOR$K_UNWIND{POP or NOP})  
163 1271 2     L_ERR_EQL_PRES: VOLATILE;    ! 1 if ERR= present  
164 1272 2  
165 1273 2 BUILTIN  
166 1274 2     ACTUALCOUNT;  
167 1275 2  
168 1276 2 ENABLE  
169 1277 2     FOR$$IOSTAT_HND (L_UNWIND_ACTION, L_ERR_EQL_PRES);  
170 1278 2     ! pass info to error handler  
171 1279 2
```

```
172 1280 2
173 1281 2
174 1282 2
175 1283 2
176 1284 2
177 1285 2
178 1286 2
179 1287 2
180 1288 2
181 1289 2
182 1290 2
183 1291 2
184 1292 2
185 1293 2
186 1294 2
187 1295 2
188 1296 2
189 1297 2
190 1298 2
191 1299 2
192 1300 2
193 1301 2
194 1302 2
195 1303 2
196 1304 2
197 1305 2
198 1306 2
199 1307 2
200 1308 2
201 1309 2
202 1310 2
203 1311 2
204 1312 2
205 1313 2
206 1314 2
207 1315 2
208 1316 2
209 1317 2
210 1318 3
211 1319 3
212 1320 3
213 1321 3
214 1322 3
215 1323 3
216 1324 3
217 1325 4
218 1326 3
219 1327 4
220 1328 4
221 1329 4
222 1330 4
223 1331 4
224 1332 5
225 1333 5
226 1334 5
227 1335 4
228 1336 3

+ Determine if ERR= is present.
-
IF ACTUALCOUNT () GTR 1
THEN
    L_ERR_EQL_PRES = .ERR_EQL
ELSE
    L_ERR_EQL_PRES = 0;

+ Set up error handler conditions in case CB_PUSH bombs
-
L_UNWIND_ACTION = FOR$K_UNWINDNOP;

+ Get a LUB for this logical unit.
+ On return, CCB points to the current control block.
-
FOR$$CB_PUSH (.UNIT, LUB$K_LUN_MIN);

+ Unwind action (if an error occurs) is now to pop a LUB.
-
L_UNWIND_ACTION = FOR$K_UNWINDPOP;

+ Check the LUB. If file is not open, then this is a no-op.
+ Else must be sequential organization and access.
-
IF .CCB [LUB$V_OPENED]
THEN
    IF NOT .CCB [LUB$V_DIRECT] AND NOT .CCB [LUB$V_NOTSEQORG]
    THEN
        BEGIN
            + Call RMS to REWIND the file, all failure codes returned
            + cause a SIGNAL_STOP to occur, except for IOP, EOF or BOF.
            -
            IF NOT (STATUS = $REWIND (RAB = .CCB))
            THEN
                BEGIN
                    IF .STATUS NEQ RMSS_IOP AND
                       .STATUS NEQ RMSS_EOF AND
                       .STATUS NEQ RMSS_BOF
                    THEN
                        BEGIN
                            FOR$$SIGNAL_STO (FOR$K_REWERR);
                            RETURN 0;
                        END;
                END;
            END;
        END;
```

229	1337
230	1338
231	1339
232	1340
233	1341
234	1342
235	1343
236	1344
237	1345
238	1346
239	1347
240	1348
241	1349
242	1350
243	1351
244	1352
245	1353
246	1354
247	1355
248	1356
249	1357
250	1358
251	1359
252	1360
253	1361
254	1362
255	1363

```

!+
!- Clear APPEND flag - OK for backspace now

CCB[LUB$V_APPEND] = 0;

!+
!- Set the logical record number to 1.

CCB[LUB$L_LOG_RECNO] = 1;

END
ELSE
BEGIN
FOR$$SIGNAL_STO (FOR$K_REWERR);
RETURN 0;
END;

!+
!- Return the file system to its former state.

FOR$$CB POP ();
RETURN 0;
END;
! Success IOSTAT value

```

```
.TITLE FOR$REWIND
.IDENT \1-007\

.EXTRN FOR$$IOSTAT_HND
.EXTRN FOR$$SIGNAL_STO
.EXTRN FOR$$CB_PUSH, FOR$$CB_POP
.EXTRN SY$$REWIND
```

.PSECT _FOR\$CODE,NOWRT, SHR, PIC,2

PC	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419
----	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

00000000G	00	5B	DD	0003D	PUSHL	CCB	:	1325
	1B	01	FB	0003F	CALLS	#1, SYSSREWIND	:	
00018574	8F	50	E8	00046	BLBS	STATUS, 3\$:	1328
		50	D1	00049	CMPL	STATUS, #99700	:	
0001827A	8F	12	13	00050	BEQL	3\$:	1329
		50	D1	00052	CMPL	STATUS, #98938	:	
00018198	8F	09	13	00059	BEQL	3\$:	1330
		50	D1	0005B	CMPL	STATUS, #98712	:	
	FD	0A	12	00062	BNEQ	4\$:	
	EO	20	8A	00064	BICB2	#32, -3(CCB)	:	1342
		01	D0	00068	MOVL	#1, -32(CCB)	:	1348
		0B	11	0006C	BRB	5\$:	1316
00000000G	00	14	DD	0006E	PUSHL	#20	:	1353
		01	FB	00070	CALLS	#1, FOR\$\$SIGNAL_STO	:	
		06	11	00077	BRB	6\$:	1354
		00	16	00079	JSB	FOR\$\$CB_POP	:	1361
		50	D4	0007F	CLRL	R0	:	1363
			04	00081	RET		:	
			0000	00082	.WORD	Save nothing	:	1263
	50	08	AC	D0 00084	MOVL	8(AP), R0	:	
	50	04	A0	D0 00088	MOVL	4(R0), R0	:	
		F8	A0	9F 0008C	PUSHAB	L_ERR_EQL PRES	:	
		FC	A0	9F 0008F	PUSHAB	L_UNWIND_ACTION	:	
		02	DD	00092	PUSHL	#2	:	
		5E	DD	00094	PUSHL	SP	:	
	7E	04	AC	7D 00096	MOVQ	4(AP), -(SP)	:	
00000000G	00	03	FB	0009A	CALLS	#3, FOR\$\$IOSTAT_HND	:	
			04	000A1	RET		:	

; Routine Size: 162 bytes, Routine Base: _FOR\$CODE + 0000

:	256	1364	1		
:	257	1365	1	END	!End of module
:	258	1366	0	ELUDOM	

PSECT SUMMARY

Name	Bytes	Attributes
_FOR\$CODE	162	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	7 0	581	00:01.3

FOR\$REWIND
1-007

L 13
16-Sep-1984 00:44:03
14-Sep-1984 12:32:40

VAX-11 BLISS-32 V4.0-742
[FORRTL.SRC]FORREWIND.B32;1

Page 7
(3)

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS:FORREWIND/OBJ=OBJ\$:FORREWIND MSRC\$:FORREWIND/UPDATE=(ENH\$:FORREWIND
:)

: Size: 162 code + 0 data bytes
: Run Time: 00:11.5
: Elapsed Time: 00:33.2
: Lines/CPU Min: 7158
: Lexemes/CPU-Min: 40302
: Memory Used: 146 pages
: Compilation Complete

0183

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY